Radical Performance Enhancements for Combinatorial Optimization Algorithms Based on the Dead-End Elimination Theorem

D. BENJAMIN GORDON,[1] STEPHEN L. MAYO[2]

[1] *Division of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, California 91125*

[2] *Howard Hughes Medical Institute and Division of Biology (147-75), California Institute of Technology, Pasadena, California 91125*

Received 10 January 1998; accepted 6 May 1998

ABSTRACT: Recent advances in protein design have demonstrated the effectiveness of optimization algorithms based on the dead-end elimination theorem. The algorithms solve the combinatorial problem of finding the optimal placement of side chains for a set of backbone coordinates. Although they are powerful tools, these algorithms have severe limitations when the number of side chain rotamers is large. This is due to the high-order time dependence of the aspect of the calculation that deals with rotamer doubles. We present three independent algorithmic enhancements that significantly increase the speed of the doubles computation. These methods work by using quantities that are inexpensive to compute as a basis for forecasting which expensive calculations are worthwhile. One of the methods, the comparison of extrema, is derived from analytical considerations, and the remaining two, the ''magic-bullet'' and the ''q_{rs}'' and ''q_{uv}'' metrics, are based on empirical observation of the distribution of energies in the system. When used together, these methods effect an overall speed improvement of as much as a factor of 47, and for the doubles aspect of the calculation, a factor of 95. Together, these enhancements extend the envelope

Correspondence to: S. L. Mayo; e-mail: steve@mayo.caltech.edu

Contract/grant sponsors: Howard Hughes Medical Institute; National Institutes of Health, contract/grant number: GM 17616C-19; Rita Allen Foundation; Chandler Family Trust; Booth Ferris Foundation; David and Lucile Packard Foundation; Searle Scholars Program; Chicago Community Trust

CCC 0192-8651 / 98 / 131505-10

of inverse folding to larger proteins by making formerly intractable calculations attainable in reasonable computer time.
J Comput Chem 19: 1505–1514, 1998

Keywords: protein design; dead-end elimination; rotamers; combinatorial optimization

Introduction

The accurate placement of side chains on a specified main chain template is of central importance to protein design and protein homology modeling. This placement is typically simplified through discretization of the conformational freedom of side chains into statistically significant representative conformations called rotamers.[1, 2] Nevertheless, the sheer number of rotameric combinations makes exhaustive searches of all arrangements computationally intractable. The dead-end elimination (DEE) theorem proposed by Desmet et al.[3] solves this problem by providing an effective means of pruning rotamers from the available combinatorial space.

Several enhancements have been proposed since the DEE theorem was first described. Fuzzy-end elimination[4] and improved forms of the elimination criteria[5] extend the utility of the theorem to more difficult problems. De Maeyer et al.[6] have demonstrated that the calculation speed can be increased by simultaneously implementing an energy threshold and a more detailed rotamer library. Together, these enhancements have enabled homology modeling calculations for proteins as large as 250 residues.

These techniques, in conjunction with systematically derived energy expressions, have been used to perform inverse protein folding (i.e., protein design). The hydrophobic cores of coiled-coil[7] and $\alpha + \beta$ proteins[8] have been successfully redesigned, as have α-helical surfaces.[9] Recently, Dahiyat and Mayo[10] have employed the DEE theorem in the complete redesign of an entire 28-residue motif.

Design calculations are significantly more computationally intensive than homology modeling calculations on proteins of the same size. This is due to a high-order time dependence on the number of allowed rotamers per residue position. Design calculations suffer because rotamers from several different amino acids are possible at each position. We present three speed enhancements that make such design calculations attainable in reasonable computer time.

Background

The strength of the DEE theorem is that it can determine that a particular rotamer cannot exist in the global minimum energy conformation (GMEC) without any prior knowledge of the GMEC. A rotamer determined to be incompatible with the GMEC is termed ''dead-ending,'' and is eliminated from further consideration. The GMEC is then attained through iterative elimination of dead-ending rotamers until only a single rotamer remains at each residue position.

To eliminate a rotamer, one must show that there exists another rotamer that contributes less energy to the GMEC than the candidate rotamer. This is accomplished by finding a rotamer that is lower in energy than the candidate in all possible configurations of the system. The DEE criterion proposed by Desmet et al.[3] confirms the lower energy for all configurations by checking if, for some residue position, i, the minimum energy of the candidate rotamer to be eliminated, i_r, is greater than the maximum energy of another rotamer, i_t:

$$E(i_r) + \sum_{j,\, j \neq i} \min_S E(i_r j_s) > E(i_t) + \sum_{j,\, j \neq i} \max_S E(i_t j_s) \quad (1)$$

The quantity $E(i_r)$ is the interaction energy of the rotamer i_r with the template. The energy of interaction between two rotamers i_r and j_s is denoted $E(i_r j_s)$. Thus, the minimum and maximum energies for all configurations are expressed on the left and right sides of the criterion, respectively.

There can be cases, however, in which the energy profile of a candidate rotamer may be higher than a reference rotamer in all conformations, but its minimum may be lower in energy than the maximum of the other rotamer. Although the can-

didate rotamer should be eliminated, the pair will be overlooked by the aforementioned elimination criterion. To treat this case, as well as higher order cases, Goldstein[5] proposed a form of the criterion of arbitrary order. The zeroth order form refers to the criterion proposed by Desmet et al. Goldstein describes a more sensitive, first-order form of the criterion that also detects the special case just described

$$E(i_r) - E(i_t) + \sum_{j,\, j\neq i} \min_{S} [\, E(i_r j_s) - E(i_t j_s)] > 0 \tag{2}$$

This criterion checks that the energy profiles of two rotamers do not cross by verifying that the minimum energy difference upon substitution of a rotamer for the candidate rotamer is always greater than zero.

In practice, the calculation typically reaches a point at which no more rotamers can be eliminated by the just-noted criterion. Lasters and Desmet[4] have described how the calculation can be continued by finding pairs of rotamers, called doubles, that cannot coexist in the GMEC. The variation is obtained by tailoring the zeroth-order criterion (1) to search for dead-ending pairs

$$\varepsilon([\, i_r j_s]) + \sum_{k,\, k\neq j\neq i} \min_{t} \varepsilon([\, i_r j_s], k_t) > \varepsilon(i_u j_v) + \sum_{k,\, k\neq j\neq i} \max_{S} \varepsilon([\, i_u j_v], k_t) \tag{3}$$

where

$$\varepsilon([\, i_r j_s]) = E(i_r) + E(j_s) + E(i_r j_s)$$

and

$$\varepsilon([\, i_r j_s], k_t) = E(i_r k_t) + E(j_s k_t)$$

The Goldstein elimination criterion can also be extended to doubles

$$\varepsilon([\, i_r j_s]) - \varepsilon([\, i_u j_v]) + \sum_{k,\, k\neq j\neq i} \min_{t} [\, \varepsilon([\, i_r j_s], k_t) - \varepsilon([\, i_u j_v], k_t)] > 0 \tag{4}$$

In contrast to singles eliminated by criteria (1) or (2), it is possible that one of the individual rotamers that constitutes a dead-ending pair may exist in the GMEC, so neither of the rotamers can be eliminated. However, as a unit, dead-ending pairs can be excluded when evaluating the minima in criteria (1) and (2) in subsequent singles calculations, enabling the elimination of more rotamers. Additionally, dead-ending pairs may be eliminated upon residue unification[11] in which a ''super-residue'' is constructed from all possible rotamer pairs for two positions. The super-residue is treated as a single residue for the remainder of the calculation.

Calculation Speed

Dead-end elimination calculations filter through enormous numbers of combinations of sequences with remarkable speed when there are few rotamers per residue position. However, calculations proceed more slowly as the number of rotamers increases. This is of primary concern in protein design applications, in which each position has rotamers from many amino acids, often totaling hundreds of rotamers. Additionally, super-residues formed through unification also contribute large numbers of rotamers. The calculation is slowed in part because more rotamers need to be eliminated. More importantly, however, is that the time to execute each iteration is significantly lengthened, because of a fourth-order dependence on the number of rotamers per residue position.

To clarify this fourth-order dependence, it is convenient to define a comparison matrix. To search exhaustively for all dead-ending rotamers at a residue position i, it is necessary to compare every rotamer to every other rotamer available at i. In the comparison matrix, each column corresponds to a particular rotamer, i_r, as a candidate for elimination, and each row corresponds to one of the possible reference rotamers i_t. If there are n rotamers at position i, then an exhaustive search of n^2-n matrix elements is necessary. Such a matrix is evaluated for each of the p positions that may be represented by i.

The computational bottlenecks, however, are the evaluation of the minimum on the left side and the maximum on the right side of the elimination criterion. The calculation of each extremum requires computation for n rotamers at each of the other residue positions j. For the zeroth-order criterion (1), the same extrema can be used repeatedly within each row or column, and therefore they need only be computed once. The calculation time therefore scales proportionally to the number of rotamers and positions, $n \times p$. However, when

the first-order criterion (2) is invoked, the minimum operator is applied to the rotamer pair, and therefore must be repeated for each matrix element. Consequently, an exhaustive search using the Goldstein variation scales as $n^2 \times p$.

The problem is exacerbated when performing doubles elimination. An i–j pair submitted to evaluation by criterion (3) will have n^2 $i_r j_s$ combinations, which will be compared with n^2 $i_u j_v$ combinations. Thus, the dimension of the comparison matrix is $n^2 \times (n^2 - 1)$, and such a matrix is constructed for each of the possible $1/2 \times p \times (p - 1)$ i–j doubles. As with zeroth-order singles, it is only necessary to evaluate the extrema once for each row and column, and so the calculation scales as $n^2 \times p^2$. However, when it becomes necessary to progress to criterion (4), a computationally expensive calculation must be performed for every matrix element. Therefore, first-order doubles iterations scale as $n^4 \times p^2$.

Performance analysis of our implementation of the DEE algorithm shows that the computation of the first-order doubles criterion (4) dominates the overall calculation time. For example, a doubles calculation with 100 rotamers at each position requires the evaluation of 10^8 matrix elements for each i–j pair. At a typical evaluation rate of 10^4 comparisons per second, the zeroth-order calculation will take p^2 seconds, but the computation of an entire matrix for first-order doubles will take p^2 hours.

Optimization

The actual number of dead-ending pairs found when using the first-order doubles criterion is much smaller than the number of comparison matrix elements. The calculation could be made much faster if there were a way to predict which matrix elements were likely to be dead-ending, which would then be confirmed with the DEE criterion. Our approach is to prejudge matrix elements by utilizing the minima and maxima precalculated for the zeroth-order calculation. For convenience, we define

$$\varepsilon_{max}([i_r j_s]) = \varepsilon([i_r j_s]) + \sum_{k \neq i \neq j} \max_t \varepsilon([i_r j_s], k_t) \quad (5)$$

$$\varepsilon_{min}([i_r j_s]) = \varepsilon([i_r j_s]) + \sum_{k \neq i \neq j} \min_t \varepsilon([i_r j_s], k_t) \quad (6)$$

$$\varepsilon_{max}([i_u j_v]) = \varepsilon([i_u j_v]) + \sum_{k \neq i \neq j} \max_t \varepsilon([i_u j_v], k_t) \quad (7)$$

$$\varepsilon_{min}([i_u j_v]) = \varepsilon([i_u j_v]) + \sum_{k \neq i \neq j} \min_t \varepsilon([i_u j_v], k_t) \quad (8)$$

These quantities are illustrated on energy profiles in Figure 1. As previously stated, the calculation of these extrema scales as n^2, rather than as n^4, because the values are the same for an entire row or column of the matrix.

It is important to emphasize that it is not necessary to discover all possible dead-ending pairs in the matrix. Although more would be preferable, it is only necessary to find a sufficient number to enable successful elimination of rotamers in the next singles iteration. It is therefore reasonable to sacrifice the discovery of some pairs to gain calculation speed.

Inspection of the energy distributions in sample matrices has revealed that an $i_u j_v$ pair that dead-end eliminates a particular $i_r j_s$ pair can also eliminate other $i_r j_s$ pairs. In fact, there are often a few $i_u j_v$ pairs, which we call ''magic bullets,'' that eliminate a significant number of $i_r j_s$ pairs. We have found that one of the most potent magic bullets is the pair for which the maximum interaction energy, $\varepsilon_{max}([i_u j_v])$, is least. We refer to this pair as $[i_u j_v]_{mb}$.

Our first speed enhancement is to replace the zeroth-order doubles calculation with first-order doubles performed only on the row of the matrix elements corresponding to the $[i_u j_v]_{mb}$ pair. The discovery of $[i_u j_v]_{mb}$ is an n^2 calculation, and the application of criterion (4) to the single row of the matrix corresponding to this rotamer pair is another n^2 calculation, so the calculation cost is comparable to a zeroth-order calculation. This magic bullet first-order calculation will discover all dead-ending pairs that would be discovered by the zeroth-order calculation. This stems from the fact that $\varepsilon_{max}([i_u j_v]_{mb})$ must be less than or equal to any $\varepsilon_{max}([i_u j_v])$ that would successfully eliminate a pair by the zeroth-order criterion.

The benefit of magic-bullet calculation is that it produces 20–60% more dead-ending pairs than the corresponding zeroth-order calculation. In practice, these extra dead-ending pairs make successive first-order singles calculations more effective, often enabling the optimization to proceed through additional iterations of singles elimination before requiring a first-order doubles calculation.

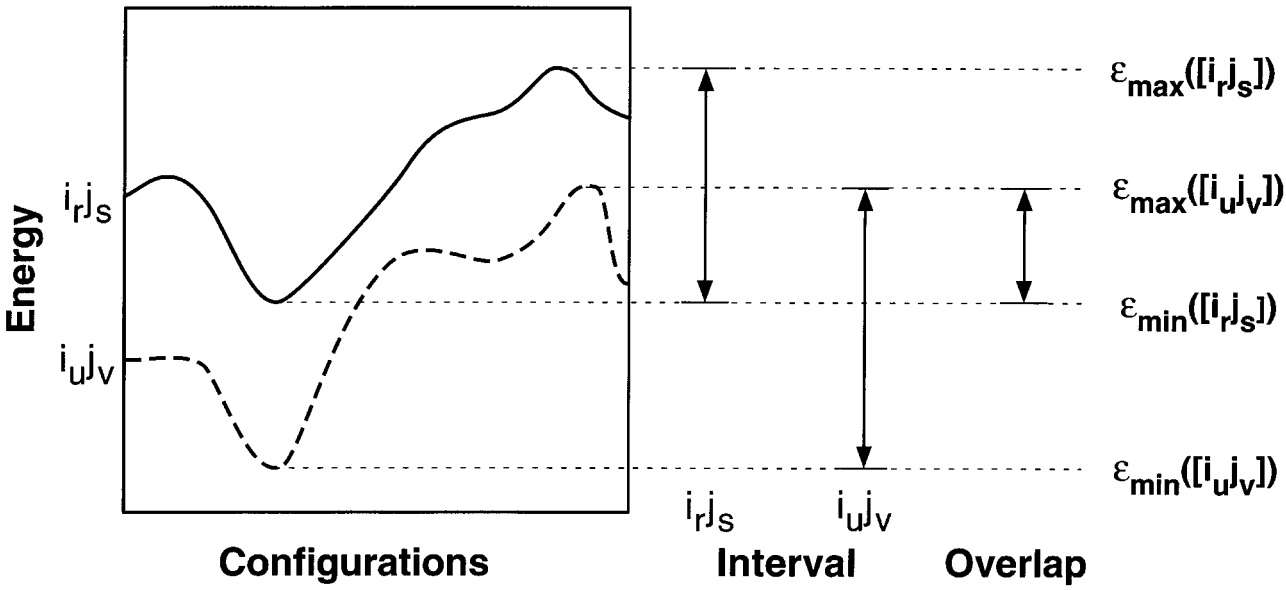

FIGURE 1. Schematic representation of the quantities defined in eqs. (5)–(8) that are used to construct speed enhancements. The minima and maxima are utilized directly to find the $[i_u j_v]_{mb}$ pair and for the comparison of extrema. The differences between the quantities, denoted with arrows, are used to construct the q_{rs} and q_{uv} metrics.

This reduces the total number of times that full first-order doubles calculations need to be performed over the course of the complete optimization.

After several iterations, the magic bullet doubles calculation fails to produce a sufficient number of dead-ending pairs, and it becomes necessary to evaluate the full doubles matrix. We observe that the remaining pairs that satisfy the first-order doubles criterion are sparse on the matrix. Therefore, many matrix elements must be searched to find a relatively small number of dead-ending pairs. The search odds can be improved, however, by using the minima and maxima precomputed earlier to isolate regions of the matrix for which the probability of finding a dead-ending pair is greater.

We employ a comparison of extrema to effectively reduce the matrix by a factor of four. Matrix elements that satisfy either of the following criteria are skipped

$$\varepsilon_{min}([i_r j_s]) < \varepsilon_{min}([i_u j_v]) \tag{9}$$

or

$$\varepsilon_{max}([i_r j_s]) < \varepsilon_{max}([i_u j_v]) \tag{10}$$

Figure 2 illustrates schematically that when either of these conditions are met, the energy profiles necessarily cross (see Appendix for proof). We can therefore be certain that the corresponding matrix element will not be dead-ending.

Because the matrix is symmetrical, half of its elements will satisfy the first inequality (9), and half of those remaining will satisfy the other inequality (10). These three quarters of the matrix need not be subjected to the evaluation of criterion (4), resulting in a theoretical speed enhancement of a factor of four.

Our last enhancement refines the search of the remaining quarter of the matrix. We accomplish this by constructing a metric from the precomputed extrema to detect those matrix elements likely to result in a dead-ending pair.

A metric was found through analysis of matrices from different sample optimizations. We searched for combinations of the extrema that predicted the likelihood that a matrix element would produce a dead-ending pair. Interval sizes (see Fig. 1) for each pair were computed from differences of the extrema. The size of the overlap of the $i_r j_s$ and $i_u j_v$ intervals were also computed, as well as the difference between the minima and the difference between the maxima. Combinations of these quantities, as well as the lone extrema, were tested for their ability to predict the occurrence of dead-ending pairs. Because some of the maxima were very large, the quantities were also compared logarithmically.

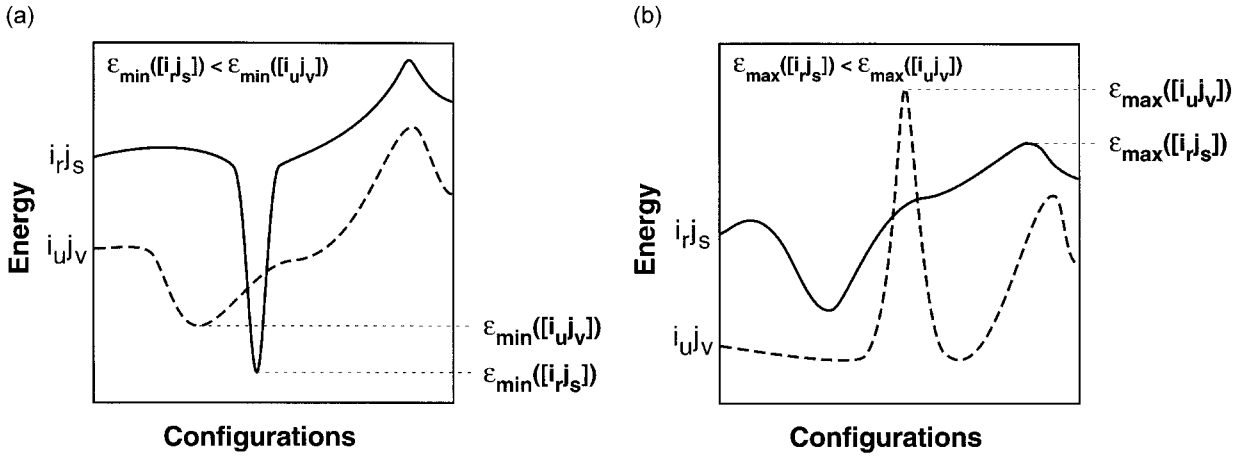

FIGURE 2. These graphs illustrate that the energy profiles of doubles must cross if they satisfy the comparison of extrema inequalities (9) in (a), and (10) in (b). One may therefore determine beforehand that these doubles cannot be eliminated by the first-order criterion.

Most of the combinations were able to predict dead-ending matrix elements to varying degrees. The best metrics were the fractional interval overlap with respect to each pair. We refer to these quotients as q_{rs} and q_{uv}

$$q_{rs} = \frac{\text{interval overlap}}{\text{interval } ([i_r j_s])} = \frac{\varepsilon_{max}([i_u j_v]) - \varepsilon_{min}([i_r j_s])}{\varepsilon_{max}([i_r j_s]) - \varepsilon_{min}([i_r j_s])} \quad (11)$$

$$q_{uv} = \frac{\text{interval overlap}}{\text{interval } ([i_u j_v])} = \frac{\varepsilon_{max}([i_u j_v]) - \varepsilon_{min}([i_r j_s])}{\varepsilon_{max}([i_u j_v]) - \varepsilon_{min}([i_u j_v])} \quad (12)$$

These metrics were selected because they yield ratios of the occurrence of dead-ending matrix elements to the total occurrence of elements that are higher than any of the other metrics we tested. For example, we observe that there are very few matrix elements ($\sim 2\%$) for which $q_{rs} > 0.98$, yet these elements produce 30–40% of all of the dead-ending pairs.

We apply the first-order doubles criterion only to those doubles for which $q_{rs} > 0.98$ and $q_{uv} < 0.99$. The sample data analyses predict that, by using these two metrics, we may find as many as half of the dead-ending elements by evaluating only 2–5% of the reduced matrix. However, we do not expect to observe the full theoretical enhancement because the analysis does not account for redundant eliminations of a pair.

Method

ENERGY EXPRESSION

The energy expression consists of van der Waals, electrostatic, and solvation terms. For van der Waals, a Lennard–Jones 6-12 potential was used, with radii scaled[8] by a factor of 0.9. A distance-dependent electrostatic term and a hybridization-dependent hydrogen-bonding term were used.[9] Solvation effects were approximated from hydrophobic surface area burial.[7, 10] Atom radii and hydrogen-bond well depths were based on the DREIDING force-field.[12]

ALGORITHM

The basic algorithm was implemented as described in the ''Background'' section of this article. Residue unification[11] was performed when first-order doubles failed to facilitate subsequent singles iterations, by clustering the pair of positions that produced the largest fraction of dead-ending pairs. Rotamers were selected from a backbone-dependent library.[13]

The three speed enhancements were added sequentially. First, calculations were performed using the original algorithm. Next, magic-bullet doubles were substituted for the zeroth-order doubles

calculation. Then a filter implementing the comparison of extrema was added to the first-order doubles calculation. Last, the q_{rs} and q_{uv} metrics were added as a final filter to the first-order doubles calculation.

For each calculation, the total CPU time was recorded, as well as the portion of that time spent performing first-order doubles. The time required for the initial first-order doubles was also measured. Calculations were performed on a single R10000 CPU of a Silicon Graphics Origin 2000 server.

BENCHMARK CASES

It was necessary to test the generality of the speed enhancements, because their viability is, in part, dependent on the distribution of energies. Therefore, three sequence optimization problems representative of different protein structural classes were selected. To test α-helical surfaces, the coiled-coil GCN4-p1[9, 14] was used. The 12 residues occupying **b**, **c**, or **f** locations in the heptad repeat were optimized by allowing each position to have the identity of any of the hydrophilic amino acids (D, E, N, Q, K, R, S, T, A, and H). There were 8.5×10^{26} rotameric combinations.

The structure of the β1 domain[15] of streptococcal protein G was used to test the applicability of the enhancements to protein cores and β-sheet surfaces. For the former, 13 positions in the core and at the boundary (3, 5, 7, 26, 30, 33, 34, 37, 43, 50, 52, 54, 56) were optimized from the 2.4×10^{23} combinations of hydrophobic rotamers (A, F, I, L, M, V, W, Y). For the β-sheet surface, 12 positions (4, 6, 8, 13, 15, 17, 42, 44, 46, 51, 53, 55) were optimized from the 1.8×10^{26} combinations of hydrophilic rotamers.

Results

The calculation times for the three benchmark cases are shown in Table I. The enhancements collectively increase the calculation speed by more than an order of magnitude. In some cases, the overall speed increase is nearly a factor of 50, and the speed enhancement for first-order doubles calculations is a factor of 95. All algorithms produce the same solutions for each optimization problem.

The evaluation times of the initial first-order doubles calculations are used as predictors of the speed enhancement for large calculations. It is not feasible to measure directly the speed enhance-

TABLE I.
Observed Speed Enhancements for Magic Bullet, Comparison of Extrema, and q_{rs} and q_{uv} Metrics on Representative Cases of Three Structural Classes.

		Total optimization time		Total first-order doubles time		Earliest first-order doubles time[e]	
	Method	Minutes	(Factor)	Minutes	(Factor)	Minutes	(Factor)
Case 1: Core and boundary	Original[a]	192.6		190.6		24.9	
	MB[b]	165.5	(×1.2)	163.7	(×1.2)	11.2	(×2.2)
	MB + CoE[c]	31.3	(×6.2)	29.8	(×6.4)	2.4	(×10)
	MB + CoE + Q[d]	14.0	(×14)	11.5	(×17)	0.5	(×52)
Case 2: Helical surface	Original	461.4		436.6		371.2	
	MB	206.5	(×2.2)	204.6	(×2.1)	152.9	(×2.4)
	MB + CoE	49.2	(×9.4)	47.3	(×9.2)	35.5	(×10)
	MB + CoE + Q	13.7	(×34)	11.4	(×38)	6.6	(×56)
Case 3: Beta-sheet surface	Original	868.6		866.1		712.7	
	MB	303.5	(×2.9)	300.8	(×2.9)	257.3	(×2.8)
	MB + CoE	71.2	(×12)	68.5	(×13)	59.1	(×12)
	MB + CoE + Q	18.4	(×47)	14.8	(×59)	7.5	(×95)

[a] The original algorithm uses the zeroth-order doubles criterion prior to evaluating the entire first-order doubles matrix.
[b] Magic bullet (MB) first-order doubles are substituted for zeroth-order doubles.
[c] The comparison of extrema (CoE) filter is employed during evaluating of the first-order doubles matrix.
[d] The metrics (Q) are used as additional filters during first-order doubles ($q_{rs} > 0.98$ and $q_{uv} < 0.99$).
[e] Execution time required for the earliest encountered iteration of first-order doubles.

ment for very large problems, due to the prohibitive calculation times for their references. Additionally, the overall enhancement is increased for calculations of large size, due to the larger fraction of the calculation dedicated to the evaluation of large doubles matrices. We therefore focus analysis on the calculation times of the earliest encountered large doubles matrix, although the trends are exhibited by the other performance measures as well.

Similar enhancements were observed for all three structural classes. The fluctuations in time improvement are apparently related to the overall difficulty of the optimizations. Harder calculations, such as those involving the weakly interacting surface residues of β-sheets, derive the greatest enhancement.

The employment of the magic bullet imparts a speed enhancement factor of 1.3–2.9, depending on the nature of the optimization problem. As desired, the enhancement enables the calculation to progress through several additional iterations before requiring the invocation of a full first-order doubles round. The size of the problem is therefore reduced for subsequent expensive doubles calculations (Table II).

The observed benefit of the comparison of extrema exceeds the theoretical enhancement for all the test cases. This is a byproduct of an implementation detail that prevents redundant eliminations. It is unnecessary to search remaining $i_u j_v$ pairs after one is found that eliminates a particular $i_r j_s$, so calculations for $i_r j_s$ pairs that are eliminated require less computation time than those that are not. The comparison of extrema filter reduces the relative number of $i_r j_s$ pairs that require comparison against all $i_u j_v$ pairs, thereby further speeding the calculation.

Last, the combination of the metrics, q_{rs} and q_{uv}, is observed to work well for the different cases, increasing the speed of the initial doubles calculation by an additional factor of 5 to 8. Coupled with the similarity of trends observed in the initial matrix analysis, we conclude that the selected metrics, q_{rs} and q_{uv}, are effective for all structural classes.

Conclusions

We have demonstrated the effectiveness of three enhancements for algorithms based on the dead-end elimination theorem. When used in concert, these techniques reduce the calculation time of the slowest parts of the algorithm by nearly two orders of magnitude in some cases. We observe that all the techniques are effective for optimizations of different protein structural classes, and that the speed enhancements increase with the difficulty of the problem.

The increase in computational speed has dramatic consequences. Previously unattainable calculations for large protein systems are now tractable in reasonable computer time.

TABLE II.
Effect of Substituting Magic-Bullet Doubles on Subsequent First-Order Doubles Calculations.

	Number of combinations at beginning of optimization	Doubles method	Number of combinations remaining when evaluating the earliest first-order doubles matrix	Average number of rotamers per position during earliest first-order doubles matrix[a]	Theoretical speed enhancement[b] (ratio)^4	Observed speed enhancement
Case 1: Core and boundary	2.4×10^{23}	Zeroth-order	9.9×10^{15}	36.2		
		magic bullet	5.9×10^{14}	30.4	2.0	2.2
Case 2: Helical surface	8.5×10^{26}	Zeroth-order	6.4×10^{21}	81.2		
		magic bullet	5.0×10^{20}	70.0	1.8	2.4
Case 3: Beta-sheet surface	1.8×10^{26}	Zeroth-order	2.3×10^{21}	80.2		
		magic bullet	7.2×10^{19}	64.1	2.4	2.8

[a]Average number of rotamers calculated by dividing the total number rotamers by the number of residue positions in the optimization.
[b]Because the evaluation time of the doubles matrix scales as (number of rotamers per position)^4, one may approximate the theoretical improvement from the relative numbers of rotamers per residue position.

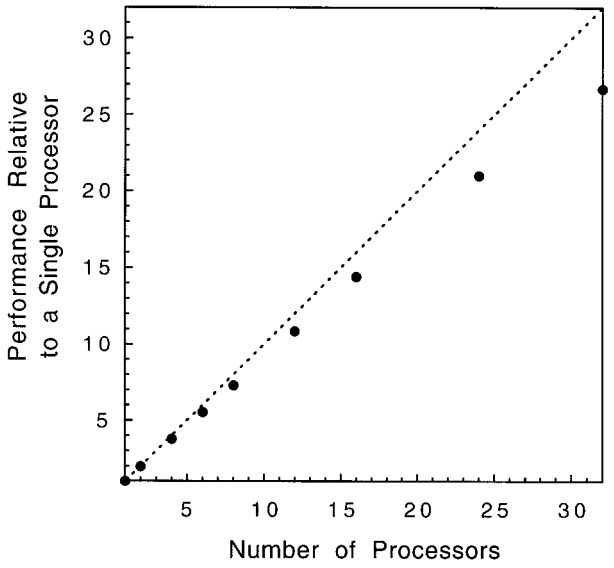

FIGURE 3. Increase in calculation speed from parallelization. Times were tabulated for a single first-order doubles iteration during the optimization of the 12 β-sheet surface positions of protein G, allowing all amino acid identities, except proline, at all positions. Performance factors are computed relative to the calculation time on a single CPU. The dotted reference line illustrates ideal performance scaling. The scaling is expected to become more ideal as the problem size increases.

Moreover, the evaluation of a large, well-defined matrix lends itself to easy computational parallelization. We have coupled these enhancements with parallelization of the doubles matrix calculation on a 32 CPU Silicon Graphics Origin 2000, and have observed that total calculation times scale nearly ideally with the number of processors used (Fig. 3). This coupling has enabled us to perform calculations in one day that previously would have taken years.

The successes of the magic-bullet and metric methods suggest that there is fertile ground in the area of optimization based on empirical observation. More sophisticated metrics may yet exist to better predict which first-order doubles calculations are worthwhile.

Acknowledgments

The authors thank A. G. Street for helpful discussions and L. S. Gordon for assistance with the mathematical proofs.

Appendix: Proof of Comparison of Extrema

To simplify the presentation, we show the proof for a singles calculation. Consider a pair of rotamers, i_r and i_t, for which we observe that

$$E_{min}(i_r) < E_{min}(i_t)$$

which means

$$E(i_r) + \sum_j \min_s E(i_r j_s) < E(i_t) + \sum_j \min_s E(i_t j_s)$$

Rearranging we obtain

$$E(i_r) - E(i_t) + \sum_j \min_s E(i_r j_s) - \sum_j \min_s E(i_t j_s) < 0$$

Let m be the selection of rotamer s for each position j that minimizes $E(i_r j_s)$. By definition, then

$$\sum_j E(i_r j_m) = \sum_j \min_s E(i_r j_s)$$

Now, because

$$\sum_j E(i_t j_m) \geq \sum_j \min_s E(i_t j_s)$$

we have

$$E(i_r) - E(i_t) + \sum_j E(i_r j_m) - \sum_j E(i_t j_m) \leq 0$$

Because m is the same in both pairwise energy expressions, we may write

$$E(i_r) - E(i_t) + \sum_j [E(i_r j_m) - (i_t j_m)] \leq 0$$

This shows that there must exist a configuration, m, for which the energy difference is negative upon substitution of i_t for i_r. Because the minimum difference upon substitution must be less than or equal to any particular difference

$$\sum_j \min_s [E(i_r j_s) - E(i_t j_s)] \leq \sum_j [E(i_r j_m) - E(i_t j_m)]$$

Substituting the minimum difference yields

$$E(i_r) - E(i_t) + \sum_j \min_s [E(i_r j_s) - E(i_t j_s)] \leq 0$$

Thus, given that the initial comparison of minima is satisfied, the minimum difference must be less than zero. Therefore, i_t cannot eliminate i_r by first-order elimination. By analogy, the same condition can be derived when the maximum energy of i_t exceeds the maximum of i_r. The proofs are analogous for doubles calculation, confirming the conditions described in the text.

References

1. J. Janin, S. Wodak, M. Levitt, and D. Maigret, *J. Mol. Biol.*, **125**, 357 (1978).
2. Ponder and F. Richards, *J. Mol. Biol.*, **193**, 775 (1987).
3. J. Desmet, M. De Maeyer, B. Hazes, and I. Lasters, *Nature*, **356**, 539 (1992).
4. I. Lasters and J. Desmet, *Prot. Eng.*, **6**, 717 (1993).
5. R. F. Goldstein, *Biophys. J.*, **66**, 1335 (1994).
6. M. De Maeyer, J. Desmet, and I. Lasters, *Folding & Design*, **2**, 53 (1997).
7. B. I. Dahiyat and S. L. Mayo, *Prot. Sci.*, **5**, 895 (1996).
8. B. I. Dahiyat and S. L. Mayo, *Proc. Natl. Acad. Sci. USA*, **94**, 10172 (1997).
9. B. I. Dahiyat, D. B. Gordon, and S. L. Mayo, *Prot. Sci.*, **6**, 1333 (1997).
10. B. I. Dahiyat and S. L. Mayo, *Science*, **278**, 82 (1997).
11. J. Desmet, M. De Maeyer, and I. Lasters, In *The Protein Folding Problem and Tertiary Structure Prediction*, K. Merz Jr. and S. Le Grand, Eds., Birkhäuser, Boston, 1994, p. 307.
12. S. L. Mayo, B. D. Olafson, and W. A. Goddard, *J. Phys. Chem.*, **94**, 8897 (1990).
13. R. L. Dunbrack and M. Karplus, *J. Mol. Biol.*, **230**, 543 (1993).
14. T. Gallagher, P. Alexander, P. Bryan, and G. L. Gilliland, *Biochemistry*, **33**, 4721 (1994).
15. E. O'Shea, J. Klemm, P. Kim, and T. Alber, *Science*, **254**, 539 (1991).